



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|-----------------|-------------|-----------------------------|---------------------|------------------|
| 10/692,004 | 10/24/2003 | Alexander J. Kolmykov-Zotov | MSFT-6146/304450.01 | 8547 |

41505 7590 03/11/2009

WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)

CIRA CENTRE, 12TH FLOOR

2929 ARCH STREET

PHILADELPHIA, PA 19104-2891

EXAMINER

JOSEPH, DENNIS P

ART UNIT

PAPER NUMBER

2629

MAIL DATE

DELIVERY MODE

03/11/2009

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/692,004

Applicant(s)

KOLMYKOV-ZOTOV ET AL.

Examiner

DENNIS P. JOSEPH

Art Unit

2629

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 20 February 2009.
2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5) ☐ Claim(s) _____ is/are allowed.
6) ☒ Claim(s) 1-20 is/are rejected.
7) ☐ Claim(s) _____ is/are objected to.
8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
10) ☒ The drawing(s) filed on 10/24/2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/CDC)
4) ☐ Interview Summary (PTO-413)
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____
Paper No(s)/Mail Date _____

DETAILED ACTION

1. This Office Action is responsive to amendments filed in application No. 10/692,004 on February 20, 2009. Claims 1-20 are pending and have been examined.

Continued Examination

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(c), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(c) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on February 20, 2009 has been entered.

Claim Rejections – 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.

4. Considering objective evidence present in the application indicating obviousness or nonobviousness.
4. **Claims 1-2, 4-10, 11-15 and 17-20** rejected under 35 U.S.C. 103(a) as being unpatentable over Morita (5,969,712) in view of Applicant's admitted prior art (AAPR), further in view of Price et al. (US 2002/0120607 A1)

Morita teaches in Claim 1:

A process for transferring pen data between unmanaged and managed code on a computing device, the unmanaged code being code native to and executed directly by a processor of the computing device (**Column 8, Lines 63-65, information processing unit to input the coordinate value (read as unmanaged code is the raw data that needs to be converted before it can be used by the application. The unit is a processor)** comprising the step of:

receiving pen data in a component on the computing device written in unmanaged code, the pen data being generated by a digitizer of the computing device upon movement of a stylus with respect to a surface of the digitizer, the pen data including at least one location on the digitizer of the stylus (**Column 8, Lines 63-65, information processing unit to input the coordinate value (read unmanaged code is the raw data that needs to be converted before it can be used by the application), used in conjunction with the coordinate detecting section 21 to indicate points. Column 13, Lines 28-31)**;

transferring information related to said pen data to a shared memory on the computing device designated to be shared between unmanaged code and managed code (**Column 9, Lines**

13-14 disclose a memory means for storing functional data, this data is based on the coordinates indicated by the user to help determine the appropriate function to select);

transferring, by the pen input component written in unmanaged code, that points to said information in said shared memory to an application on the computing device written at least in part in managed code (**Column 9, Lines 14-21 disclose a conversion means and Column 22, Lines 40-43 disclose the data is converted in accordance to perform the assigned function, (read as being converted by a means to now be partly in managed code form to allow the function to be executed); and**

retrieving, said information from said shared memory (**Column 9, Lines 13-14 disclose a memory means for manipulating the functional data); but**

However, Morita does not explicitly teach of the two types of codes “being executed in a common language run-time environment of a framework operating on the computing device, the common language run-time environment of the framework executing the managed code independent of a type of the processor of the computing device.” Specifically, Morita does not teach of the CLR. However, such a framework is well known in the art.

To emphasize, the prior art provided by Applicant, Figure 3 shows the common language run-time to manage the various types of codes, such as managed and unmanaged code. [0009] of the disclosure states that one may be motivated to use the CLR to transfer information (unmanaged code) to a managed application. Furthermore, Applicant’s prior art shows of a pen services component 302 (read as pen input component) and also discloses a type library importer 306.

Part of this importer can import stylus packets manually into managed space (read as it being managed code), so this is can reasonably be interpreted as a stylus input subsystem and is separate from the pen services component. This is part of the CLR that is admitted by Applicant.

Therefore, it would be obvious to one of ordinary skill in the art at the time of the invention to use the common language run-time environment of a framework to receive the coordinate data (unmanaged code) of Morita's stylus input and to use its environment to convert the code for use with the appropriate application (managed code). The CLR is well known and is provided as prior art by the Applicant. The motivation is that is commonly used in the art and Applicant's disclosure makes for it to be enabling.

Morita also does not explicitly teach of a pointer to reference different parts of the memory. However, pointers are well known for this function, to point to addresses for retrieval. Examiner takes Official Notice as to the use of these as they are well known in the art.

Morita also does not explicitly teach of a "mutual exclusion" shared memory so that non-simultaneous sharing of code can be achieved between the unmanaged and managed sections.

However, the use of mutex is well known in the art.

To emphasize, Price teaches of using a mutex to allow multiple programs to share the same resource, but not simultaneously, (Price, [0060]).

Therefore, it would be obvious to one of ordinary skill in the art at the time of the invention to integrate the mutex, as taught by Price, with Morita's device, as modified by AAPA, with the motivation that by non-simultaneously sharing the mutex, sequential operation can be done and errors can be prevented from different programs trying to access the same object at the same time.

Morita teaches in Claim 2:

The process according to claim 1, further comprising the steps of:

transferring additional information from said at least in part managed application to said shared memory (**Abstract, conversion status output means 25 outputs data stored in the memory means 23, for data that is converted**);

transferring a pointer that points to said additional information to said component
(**Column 9, Lines 13-14 disclose a memory which can be accessed**);
retrieving said additional information from said shared memory. (**Column 9, Lines 13-14 disclose a memory which can be accessed for functional data, repetitive for multiple stylus inputs**)

Morita teaches in Claim 4:

The process according to claim 1, further comprising the step of: exchanging information through a COM interface. (**Column 8, Lines 63-65, information processing unit to input the coordinate value, used in conjunction with the coordinate detecting section 21 to indicate points. Column 13, Lines 28-31. This interacts with the stylus to determine the inputs**)

Morita teaches in Claim 5:

The process according to claim 1, said component being a pen services component.

(Column 13, Lines 28-31 disclose the coordinate detecting section 21 to interact with the stylus/pen)

Morita teaches in Claim 6:

The process according to claim 1, said application including a pen input managed client.

(Column 8, Lines 63-65, information processing unit to input the coordinate value)

Morita teaches in Claim 7:

The process according to claim 1, said component receiving input from at least one pen device driver. **(Figure 2 shows a stylus 3 being used to input coordinates)**

Morita teaches in Claim 8:

A system for transferring information between unmanaged code and managed code on a computing device, the unmanaged code being code native to and executed directly by a processor of the computing device **(Column 8, Lines 63-65, information processing unit to input the coordinate value (read as unmanaged code is the raw data that needs to be converted before it can be used by the application. The unit is a processor)** comprising:

a shared memory on the computing device designated to be shared between unmanaged code and managed code **(Column 9, Lines 13-14 disclose a memory means for storing**

functional data, this data is based on the coordinates indicated by the user to help determine the appropriate function to select)

a component on the computing device that is written in unmanaged code (**coordinate data**) and that receives pen data, the pen data being generated by a digitizer of the computing device upon movement of a stylus with respect to a surface of the digitizer, the pen data including at least one location on the digitizer of the stylus, the component transferring information relating to said pen data to said shared memory and transfers points to said information in the shared memory to an application on the computing device having managed code (**Column 8, Lines 63-65, information processing unit to input the coordinate value (read unmanaged code is the raw data that needs to be converted before it can be used by the application), used in conjunction with the coordinate detecting section 21 to indicate points. Column 13, Lines 28-31. Column 9, Lines 14-21 disclose a conversion means and Column 22, Lines 40-43 disclose the data is converted in accordance to perform the assigned function, (read as being converted by a means to now be partly in managed code form to allow for the function to be executed);**

said application having managed code receiving said pointer and obtaining said information from said shared memory. (**Column 9, Lines 13-14 disclose a memory means for manipulating the functional data**)

However, Morita does not explicitly teach of the two types of codes “being executed in a common language run-time environment of a framework operating on the computing device, the common language run-time environment of the framework executing the managed code

independent of a type of the processor of the computing device.” Specifically, Morita does not teach of the CLR. However, such a framework is well known in the art.

To emphasize, the prior art provided by Applicant, Figure 3 shows the common language run-time to manage the various types of codes, such as managed and unmanaged code. [0009] of the disclosure states that one may be motivated to use the CLR to transfer information (unmanaged code) to a managed application. Furthermore, Applicant’s prior art shows of a pen services component 302 (read as pen input component) and also discloses a type library importer 306. Part of this importer can import stylus packets manually into managed space (read as it being managed code), so this is can reasonably be interpreted as a stylus input subsystem and is separate from the pen services component. This is part of the CLR that is admitted by Applicant.

Therefore, it would be obvious to one of ordinary skill in the art at the time of the invention to use the common language run-time environment of a framework to receive the coordinate data (unmanaged code) of Morita’s stylus input and to use its environment to convert the code for use with the appropriate application (managed code). The CLR is well known and is provided as prior art by the Applicant. The motivation is that is commonly used in the art and Applicant’s disclosure makes for it to be enabling.

Morita also does not explicitly teach of a pointer to reference different parts of the memory. However, pointers are well known for this function, to point to addresses for retrieval. Examiner takes Official Notice as to the use of these as they are well known in the art.

Morita also does not explicitly teach of a “mutual exclusion” shared memory so that non-simultaneous sharing of code can be achieved between the unmanaged and managed sections.

However, the use of mutex is well known in the art.

To emphasize, Price teaches of using a mutex to allow multiple programs to share the same resource, but not simultaneously, (Price, [0060]).

Therefore, it would be obvious to one of ordinary skill in the art at the time of the invention to integrate the mutex, as taught by Price, with Morita's device, as modified by AAPA, with the motivation that by non-simultaneously sharing the mutex, sequential operation can be done and errors can be prevented from different programs trying to access the same object at the same time.

Morita teaches in Claim 9:

The system according to claim 8, said component exposing a COM interface. (**Column 8, Lines 63-65, information processing unit to input the coordinate value, used in conjunction with the coordinate detecting section 21 to indicate points. Column 13, Lines 28-31. This interacts with the stylus to determine the inputs**)

Morita teaches in Claim 11:

The system according to claim 8, said component including a pen services component.
(Column 13, Lines 28-31 disclose the coordinate detecting section 21 to interact with the stylus/pen)

Morita teaches in Claim 12:

The system according to claim 8, further comprising: at least one pen device driver sending information to said component. **(Figure 2 shows a stylus 3 being used to input coordinates)**

Morita teaches in Claim 13:

The system according to claim 8, further comprising: said application including a pen input managed client. **(Column 8, Lines 63-65, information processing unit to input the coordinate value)**

Morita teaches in Claim 14:

A computer-readable storage medium having a program stored thereon for transferring information related to ink between an unmanaged pen input component and an application on a computing device, the unmanaged code being code native to and executed directly by a processor of the computing device **(Column 8, Lines 63-65, information processing unit to input the coordinate value (read as unmanaged code is the raw data that needs to be converted before it can be used by the application. The unit is a processor), said program comprising the steps of (Figure 2 shows a stylus input for a device):**

receiving pen data in the unmanaged component on the computing device, the pen data being generated by a digitizer of the computing device upon movement of a stylus with respect to a surface of the digitizer, the pen data including at least one location on the digitizer of the stylus (**Column 8, Lines 63-65, information processing unit to input the coordinate value (read unmanaged code as raw data that needs to be converted before it can be used by the application), used in conjunction with the coordinate detecting section 21 to indicate points. Column 13, Lines 28-31);**

transferring information related to said pen data to a shared memory on the computing device designated to be shared between unmanaged code and managed code (**Column 9, Lines 13-14 disclose a memory means for storing functional data, this data is based on the coordinates indicated by the user to help determine the appropriate function to select);**

transferring, by the unmanaged component, points to said information in said shared memory to an application written at least in part in managed code (**Column 9, Lines 14-21 disclose a conversion means and Column 22, Lines 40-43 disclose the data is converted in accordance to perform the assigned function, (read as being converted by a means to now be partly in managed code form to allow the function to be executed); and**

retrieving said information from said shared memory. (**Column 9, Lines 13-14 disclose a memory means for manipulating the functional data)**

However, Morita does not explicitly teach of the two types of codes “being executed in a common language run-time environment of a framework operating on the computing device, the common language run-time environment of the framework executing the managed code

independent of a type of the processor of the computing device.” Specifically, Morita does not teach of the CLR. However, such a framework is well known in the art.

To emphasize, the prior art provided by Applicant, Figure 3 shows the common language run-time to manage the various types of codes, such as managed and unmanaged code. [0009] of the disclosure states that one may be motivated to use the CLR to transfer information (unmanaged code) to a managed application. Furthermore, Applicant’s prior art shows of a pen services component 302 (read as pen input component) and also discloses a type library importer 306. Part of this importer can import stylus packets manually into managed space (read as it being managed code), so this is can reasonably be interpreted as a stylus input subsystem and is separate from the pen services component. This is part of the CLR that is admitted by Applicant.

Therefore, it would be obvious to one of ordinary skill in the art at the time of the invention to use the common language run-time environment of a framework to receive the coordinate data (unmanaged code) of Morita’s stylus input and to use its environment to convert the code for use with the appropriate application (managed code). The CLR is well known and is provided as prior art by the Applicant. The motivation is that is commonly used in the art and Applicant’s disclosure makes for it to be enabling.

Morita also does not explicitly teach of a pointer to reference different parts of the memory. However, pointers are well known for this function, to point to addresses for retrieval. Examiner takes Official Notice as to the use of these as they are well known in the art.

Morita also does not explicitly teach of a “mutual exclusion” shared memory so that non-simultaneous sharing of code can be achieved between the unmanaged and managed sections.

However, the use of mutex is well known in the art.

To emphasize, Price teaches of using a mutex to allow multiple programs to share the same resource, but not simultaneously, (Price, [0060]).

Therefore, it would be obvious to one of ordinary skill in the art at the time of the invention to integrate the mutex, as taught by Price, with Morita's device, as modified by AAPA, with the motivation that by non-simultaneously sharing the mutex, sequential operation can be done and errors can be prevented from different programs trying to access the same object at the same time.

Morita teaches in Claim 15:

The computer-readable storage medium according to claim 14, said program further comprising the steps of:

transferring additional information from said at least in part managed application to said shared memory (**Abstract, conversion status output means 25 outputs data stored in the memory means 23, for data that is converted**);

transferring a pointer that points to said additional information to said component
(**Column 9, Lines 13-14 disclose a memory which can be accessed**);
retrieving said additional information from said shared memory. (**Column 9, Lines 13-14 disclose a memory means for manipulating the functional data**)

Morita teaches in Claim 17:

The computer-readable storage medium according to claim 14, said program further comprising the step of: exchanging information through a COM interface. (**Column 8, Lines 63-65, information processing unit to input the coordinate value. Used in conjunction with the coordinate detecting section 21 to indicate points. Column 13, Lines 28-31. This interacts with the stylus to determine the inputs**)

Morita teaches in Claim 18:

The computer-readable storage medium according to claim 14, said component being a pen services component. (**Column 13, Lines 28-31 disclose the coordinate detecting section 21 to interact with the stylus/pen**)

Morita teaches in Claim 19:

The computer-readable storage medium according to claim 14, said application including a pen input managed client. (**Column 8, Lines 63-65, information processing unit to input the coordinate value**)

Morita teaches in Claim 20:

The computer-readable storage medium according to claim 14, said component receiving input from at least one pen device driver. **(Figure 2 shows a stylus 3 being used to input coordinates)**

As per Claims 3, 10 and 16:

These claims are directed to the use of a P-invoke style interface. Examiner takes Official Notice as to the use of P-invoke interfacing. This is common in the art as a software means.

Response to Arguments

Applicant's arguments considered, but are in moot in view of the new rejection.

Applicant is thanked for the telephone conversation for discussing the previous rejection and how to overcome it. The differences between the prior art Figure 3 and the present invention of Figure 4 were discussed and Applicant was advised to better claim the differences between the two figures.

While Examiner realizes that steps are being taken to do this, such as with the addition of the stylus input subsystem, the language is still broad enough to be read on by the prior art. What is now being claimed is the stylus input subsystem, separate from the pen input component and being written in managed code. However, Applicant's background states that the library importer 306 of Figure 3 can important the stylus packets into managed space. This is taking the information from the pen services component 302, separate from 306. As currently claimed, Applicant has patentably claimed that there are two types of input components now, but the prior

art still shows this. It is important to note that the functionality of the stylus input subsystem has not been described in any way from the claim language, just that it is separate from the pen input component and retrieves information from the latter. Since no functionality is claimed, it is difficult to differentiate it from the prior art.

It seems Applicant's background does not go into much detail on the importing of the stylus information, nor does he show much in the prior art. As a result, Applicant is advised to claim functionality for the stylus input subsystem to help overcome the current rejection. Again, just claiming the existence of it, which in Examiner's opinion, is all that is being done, is not enough to overcome a modified interpretation of the current rejection.

As for the mutual exclusion shared memory, mutex are common in the art and are used to buffer against programs accessing the same object simultaneously. Price has been combined to teach of this.

Conclusions

Any inquiry concerning this communication or earlier communications from the examiner should be directed to DENNIS P. JOSEPH whose telephone number is (571)270-1459. The examiner can normally be reached on Monday-Friday, 8am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Amr Awad can be reached on 571-272-7764. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

DJ

/Amr Awad/
Supervisory Patent Examiner, Art Unit 2629